



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/647,058	08/21/2003	J. Patrick Thompson	MSFT-1748/302722.01	1588
41505	7590	05/03/2006	EXAMINER	
WOODCOCK WASHBURN LLP (MICROSOFT CORPORATION)			VAUTROT, DENNIS L	
ONE LIBERTY PLACE - 46TH FLOOR			ART UNIT	
PHILADELPHIA, PA 19103			PAPER NUMBER	

2167

DATE MAILED: 05/03/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 10/647,058	Applicant(s) THOMPSON ET AL.	
	Examiner Dennis L. Vautrot	Art Unit 2167	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 21 August 2003.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-60 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-60 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 21 August 2003 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|---|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date <u>2/9/2004</u> . | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

Information Disclosure Statement

1. The information disclosure statement (IDS) submitted on 9 February 2004 has been received and entered into the record. Since the IDS complies with the provisions of MPEP § 609, the references cited therein have been considered by the examiner. See attached forms PTO-1449.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

3. Claims 1-3, 5, 7-8, 10-11, 21-22, 26-28, 30, 37-39, 41, 43-45, 47, 49-51, 53, 55-57 and 59 are rejected under 35 U.S.C. 102(e) as being anticipated by **Anglin et al.** (US Patent Application Publication 2004/0199521).

4. Regarding claim 1, **Anglin et al.** teaches a computer system comprising: a plurality of Items comprising at least one Item, where each of said plurality of Items

Art Unit: 2167

constitute a discrete storable unit of information that can be manipulated by a hardware/software interface system (See page 2, paragraph [0020] "FIG. 2a illustrates the data structure of a storage object entry or record in the storage database that is added whenever a storage object is confirmed as written to the storage. The entry includes a unique identifier that uniquely identifies the storage object and entry in the storage database..." In **Anglin et al.**, the "Item" of the claim is referred to as a 'storage object'); a plurality of Item Folders comprising at least one Item Folder, wherein said plurality of Item Folders constitute an organizational structure for said Items (See page 2, paragraph [0024] "Storage objects may be defined as part of a storage group by adding the group identified of the one or more groups in which the storage object is a member to the group field of the storage object entry." In **Anglin et al.**, the "Item Folder" of the claim is referred to as a "storage group"); and a hardware/software interface system for manipulating said plurality of Items, wherein each of said plurality of Items belongs to at least one of said plurality of Item Folders, and wherein each of said plurality of Items may belong to more than one Item Folder of said plurality of Item Folders (See page 2, paragraph [0018] "The Server includes a storage management server program that is capable of performing storage related operations of data objects received from data management client programs. The storage management operations may comprise backup operations, archival operations, hierarchical storage management related operations or any type of storage management operations known in the art..." and see page 2, paragraph [0019] "If the storage objects are associated with an object group, then indication of the associated object group would be included

Art Unit: 2167

with the object information in the storage database...” and see page 2, paragraph

[0024] “...by adding the group identifier of the one or more groups in which the storage object is a member...” – In other words, the “Item” [“storage object”] may belong to more than one folder [“storage group”]).

5. Regarding claim 2, **Anglin et al.** teaches an Item is a member of an Item Folder but is not owned by said Item Folder, such that the deletion of said Item Folder does not automatically result in the deletion of said Item. (See page 3, paragraph [0028] “The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i.” This follows the request to delete the “group leader” which represents the group ID of the “storage group” or the “Item Folder” as in referred to in the claim.)

6. Regarding claim 3, **Anglin et al.** teaches an item is automatically deleted when it no longer belongs to any Item Folder. (See page 3, paragraph [0029] “If the target group is the only indicated group in the associated groups fields for the specified storage object entry, then the storage management server deletes the specified storage object entry from the storage database and deletes the identifier of the deleted storage object entry from the group entry for the target group.”)

Art Unit: 2167

7. Regarding claim 5, **Anglin et al.** teaches said Item is automatically deleted when it is a member of only one Item Folder and said Item Folder is deleted. (See page 3, paragraph [0028] “The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i.” This follows the request to delete the “group leader” which represents the group ID of the “storage group” or the “Item Folder” as in referred to in the claim.)

8. Regarding claim 7, **Anglin et al.** teaches each Item is a member of at least one Item Folder but is not owned by said Item Folder, such that the deletion of said Item Folder does not automatically result in the deletion of an Item. (See page 3, paragraph [0028] “The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i.” This follows the request to delete the “group leader” which represents the group ID of the “storage group” or the “Item Folder” as in referred to in the claim.)

9. Regarding claim 8, **Anglin et al.** teaches each said Item is itself automatically deleted when it no longer belongs to any Item Folder. (See page 3, paragraph [0029] “If the target group is the only indicated group in the associated groups fields for the specified storage object entry, then the storage management server deletes the

Art Unit: 2167

specified storage object entry from the storage database and deletes the identifier of the deleted storage object entry from the group entry for the target group.”)

10. Regarding claim 10, **Anglin et al.** teaches each said Item is itself automatically deleted when it is a member of only one Item Folder and said Item Folder is deleted. (See page 3, paragraph [0028] “The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i.” This follows the request to delete the “group leader” which represents the group ID of the “storage group” or the “Item Folder” as in referred to in the claim.)

11. Regarding claim 21, **Anglin et al.** teaches a hardware/software interface system capable of manipulating an Item (See page 2, paragraph [0018] “The Server includes a storage management server program that is capable of performing storage related operations of data objects received from data management client programs. The storage management operations may comprise backup operations, archival operations, hierarchical storage management related operations or any type of storage management operations known in the art...”); said Item comprising a discrete unit of information comprising a basic set of properties commonly supported across objects exposed by an operating system shell. (See page 2, paragraph [0020] “FIG. 2a illustrates the data structure of a storage object entry or record in the storage database

that is added whenever a storage object is confirmed as written to the storage. The entry includes a unique identifier that uniquely identifies the storage object and entry in the storage database...” In **Anglin et al.**, the “Item” of the claim is referred to as a “storage object”).

12. Regarding claim 22, **Anglin et al.** teaches said Item is a fundamental unit of information manipulated by an operating system. (See page 5, paragraph [0039] “...was maintained in a server that provided access for multiple clients to a file system in the server.” This is a commonly known function of an operating system.).

13. Regarding claim 26, **Anglin et al.** teaches said Item is a member of an Item Folder (see page 2, paragraph [0019] “If the storage objects are associated with an object group, then indication of the associated object group would be included with the object information in the storage database...” In the reference, the “Item” of the instant application is called the “storage object” and the “Item folder” of the instant application is called the “object group” or in some cases the “storage group”).

14. Regarding claim 27, **Anglin et al.** teaches said Item is not owned by said Item Folder, such that the deletion of said Item Folder does not automatically result in the deletion of said Item. (See page 3, paragraph [0028] “The secondary deletion ensures that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship

Art Unit: 2167

between the storage object and group i.” This follows the request to delete the “group leader” which represents the group ID of the “storage group” or the “Item Folder” as in referred to in the claim.)

15. Regarding claim 28, **Anglin et al.** teaches said item is automatically deleted when it no longer belongs to any Item Folder. (See page 3, paragraph [0029] “If the target group is the only indicated group in the associated groups fields for the specified storage object entry, then the storage management server deletes the specified storage object entry from the storage database and deletes the identifier of the deleted storage object entry from the group entry for the target group.”)

16. Regarding claim 30, **Anglin et al.** teaches said Item is automatically deleted when it is a member of only one Item Folder and said Item Folder is deleted. (See page 3, paragraph [0028] “The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i.” This follows the request to delete the “group leader” which represents the group ID of the “storage group” or the “Item Folder” as in referred to in the claim.)

17. Regarding claim 37, **Anglin et al.** teaches a method for organizing Items in computer system, said Items comprising a discrete unit of information that can be

manipulated by a hardware/software interface system (See page 2, paragraph [0018] “The Server includes a storage management server program that is capable of performing storage related operations of data objects received from data management client programs. The storage management operations may comprise backup operations, archival operations, hierarchical storage management related operations or any type of storage management operations known in the art...”); said method comprising means by which an Item can be a member of at least two Item Folders (See page 3, paragraph [0026] “Many type of storage object management operations would have to take into account group characteristics and that a storage object may be a member of multiple group types.”); but is not owned by any of said Item Folders such that the deletion of any of said Item Folders does not automatically result in the deletion of said Item (See page 3, paragraph [0028] “The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i.” This follows the request to delete the “group leader” which represents the group ID of the “storage group” or the “Item Folder” as in referred to in the claim.)

18. Regarding claim 38, **Anglin et al.** teaches the Item is a member of an Item Folder but is not owned by said Item Folder, such that the deletion of said Item Folder does not automatically result in the deletion of said Item. (See page 3, paragraph [0028] “The secondary deletion ensure that a storage object and corresponding storage

Art Unit: 2167

object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i.” This follows the request to delete the “group leader” which represents the group ID of the “storage group” or the “Item Folder” as in referred to in the claim.)

19. Regarding claim 39, **Anglin et al.** teaches the item is automatically deleted when it no longer belongs to any Item Folder. (See page 3, paragraph [0029] “If the target group is the only indicated group in the associated groups fields for the specified storage object entry, then the storage management server deletes the specified storage object entry from the storage database and deletes the identifier of the deleted storage object entry from the group entry for the target group.”)

20. Regarding claim 41, **Anglin et al.** teaches the Item is automatically deleted when it is a member of only one Item Folder and said Item Folder is deleted. (See page 3, paragraph [0028] “The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i.” This follows the request to delete the “group leader” which represents the group ID of the “storage group” or the “Item Folder” as in referred to in the claim.)

Art Unit: 2167

21. Regarding claim 43, **Anglin et al.** teaches a computer-readable medium comprising computer-readable instructions for an Item, said Items comprising a discrete unit of information that can be manipulated by a hardware/software interface system (See page 2, paragraph [0018] "The Server includes a storage management server program that is capable of performing storage related operations of data objects received from data management client programs. The storage management operations may comprise backup operations, archival operations, hierarchical storage management related operations or any type of storage management operations known in the art...")

22. Regarding claim 44, **Anglin et al.** teaches the Item is a member of an Item Folder but is not owned by said Item Folder, such that the deletion of said Item Folder does not automatically result in the deletion of said Item. (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

23. Regarding claim 45, **Anglin et al.** teaches the item is automatically deleted when it no longer belongs to any Item Folder. (See page 3, paragraph [0029] "If the target group is the only indicated group in the associated groups fields for the specified

Art Unit: 2167

storage object entry, then the storage management server deletes the specified storage object entry from the storage database and deletes the identifier of the deleted storage object entry from the group entry for the target group.”)

24. Regarding claim 47, **Anglin et al.** teaches the Item is automatically deleted when it is a member of only one Item Folder and said Item Folder is deleted. (See page 3, paragraph [0028] “The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i.” This follows the request to delete the “group leader” which represents the group ID of the “storage group” or the “Item Folder” as in referred to in the claim.)

25. Regarding claim 49, **Anglin et al.** teaches a computer readable medium comprising computer-readable instructions for organizing Items in computer system, said Items comprising a discrete unit of information that can be manipulated by a hardware/software interface system (See page 2, paragraph [0018] “The Server includes a storage management server program that is capable of performing storage related operations of data objects received from data management client programs. The storage management operations may comprise backup operations, archival operations, hierarchical storage management related operations or any type of storage management operations known in the art...”); said method comprising means by which

Art Unit: 2167

an Item can be a member of at least two Item Folders (See page 3, paragraph [0026] "Many type of storage object management operations would have to take into account group characteristics and that a storage object may be a member of multiple group types."); but is not owned by any of said Item Folders such that the deletion of any of said Item Folders does not automatically result in the deletion of said Item (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

26. Regarding claim 50, **Anglin et al.** teaches the Item is a member of an Item Folder but is not owned by said Item Folder, such that the deletion of said Item Folder does not automatically result in the deletion of said Item. (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

Art Unit: 2167

27. Regarding claim 51, **Anglin et al.** teaches the item is automatically deleted when it no longer belongs to any Item Folder. (See page 3, paragraph [0029] “If the target group is the only indicated group in the associated groups fields for the specified storage object entry, then the storage management server deletes the specified storage object entry from the storage database and deletes the identifier of the deleted storage object entry from the group entry for the target group.”)

28. Regarding claim 53, **Anglin et al.** teaches the Item is automatically deleted when it is a member of only one Item Folder and said Item Folder is deleted. (See page 3, paragraph [0028] “The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i.” This follows the request to delete the “group leader” which represents the group ID of the “storage group” or the “Item Folder” as in referred to in the claim.)

29. Regarding claim 55, **Anglin et al.** teaches a computer-readable medium comprising computer-readable instructions for a hardware/software interface system, said operating system comprising: means for manipulating a plurality of Items comprising at least one Item, where each of said plurality of Items constitute a discrete unit of information that can be manipulated by a hardware/software interface system (See page 2, paragraph [0020] “FIG. 2a illustrates the data structure of a storage object

Art Unit: 2167

entry or record in the storage database that is added whenever a storage object is confirmed as written to the storage. The entry includes a unique identifier that uniquely identifies the storage object and entry in the storage database...” In **Anglin et al.**, the “Item” of the claim is referred to as a “storage object”); means for manipulating a plurality of Item Folders comprising at least one Item Folder, wherein said plurality of Item Folders constitute an organizational structure for said Items (See page 2, paragraph [0024] “Storage objects may be defined as part of a storage group by adding the group identifier of the one or more groups in which the storage object is a member to the group field of the storage object entry.” In **Anglin et al.**, the “Item Folder” of the claim is referred to as a “storage group”); and wherein each of said plurality of Items belongs to at least one of said plurality of Item Folders, and wherein each of said plurality of Items may belong to more than one Item Folder of said plurality of Item Folders (See page 2, paragraph [0018] “The Server includes a storage management server program that is capable of performing storage related operations of data objects received from data management client programs. The storage management operations may comprise backup operations, archival operations, hierarchical storage management related operations or any type of storage management operations known in the art...” and see page 2, paragraph [0019] “If the storage objects are associated with an object group, then indication of the associated object group would be included with the object information in the storage database...” and see page 2, paragraph [0024] “...by adding the group identifier of the one or more groups in which the storage

Art Unit: 2167

object is a member..." – In other words, the "Item" ["storage object"] may belong to more than one folder ["storage group"].

30. Regarding claim 56, **Anglin et al.** teaches an Item is a member of an Item Folder but is not owned by said Item Folder, such that the deletion of said Item Folder does not automatically result in the deletion of said Item. (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

31. Regarding claim 57, **Anglin et al.** teaches an item is automatically deleted when it no longer belongs to any Item Folder. (See page 3, paragraph [0029] "If the target group is the only indicated group in the associated groups fields for the specified storage object entry, then the storage management server deletes the specified storage object entry from the storage database and deletes the identifier of the deleted storage object entry from the group entry for the target group.")

32. Regarding claim 59, **Anglin et al.** teaches said Item is automatically deleted when it is a member of only one Item Folder and said Item Folder is deleted. (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and

Art Unit: 2167

corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i.” This follows the request to delete the “group leader” which represents the group ID of the “storage group” or the “Item Folder” as in referred to in the claim.)

Claim Rejections - 35 USC § 103

33. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

34. Claims 4, 6, 9 and 11 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Anglin et al.** as applied to claim 1 above, and further in view of **Edwards** (US Patent Application Publication 2004/0072560).

35. Regarding claim 4, **Anglin et al.** teaches a computer system substantially as claimed. **Anglin et al.** fails to teach an Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder. However, **Edwards** teaches an Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] “The synchroniser can be set to ‘Recycle’ rather than delete files. This means that whenever the synchroniser

is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of **Anglin et al.** with the disclosure of a recycling method of **Edwards** by simply adding the recycling method of **Edwards** to the system of **Anglin et al.** **Edwards** points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have an Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder.

36. Regarding claim 6, **Anglin et al.** teaches a computer system substantially as claimed. **Anglin et al.** fails to teach said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder. However, **Edwards** teaches said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the

Art Unit: 2167

default Item Folder.) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of **Anglin et al.** with the disclosure of a recycling method of **Edwards** by simply adding the recycling method of **Edwards** to the system of **Anglin et al.** **Edwards** points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder.

37. Regarding claim 9, **Anglin et al.** teaches a computer system substantially as claimed. **Anglin et al.** fails to teach each said Item, when each no longer belongs to any Item Folder, automatically become members of a default Item Folder. However, **Edwards** teaches each said Item, when each no longer belongs to any Item Folder, automatically become members of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder.) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of **Anglin et al.** with the disclosure of a recycling method of **Edwards** by simply adding the recycling method of **Edwards** to the system of **Anglin et al.** **Edwards** points out the

Art Unit: 2167

advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have each said Item, when each no longer belongs to any Item Folder, automatically become members of a default Item Folder.

38. Regarding claim 11, **Anglin et al.** teaches a computer system substantially as claimed. **Anglin et al.** fails to teach each said Item, when each is a member of only one Item Folder and said Item Folder is deleted, automatically become members of a default Item Folder. However, **Edwards** teaches each said Item, when each is a member of only one Item Folder and said Item Folder is deleted, automatically become members of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder.) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of **Anglin et al.** with the disclosure of a recycling method of **Edwards** by simply adding the recycling method of **Edwards** to the system of **Anglin et al.** **Edwards** points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have

been motivated to have each said Item, when each is a member of only one Item Folder and said Item Folder is deleted, automatically become members of a default Item Folder.

39. Claims 12-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Anglin et al.** as applied to claim 1 above, and further in view of **Chang et al.** (US 6,578,046).

40. Regarding claim 12, **Anglin et al.** teaches a computer system substantially as shown. **Anglin et al.** fails to teach a plurality of Categories comprising at least one Category, wherein said plurality of Categories constitute an organizational structure for said Items. However, **Chang et al.** teaches a plurality of Categories comprising at least one Category, wherein said plurality of Categories constitute an organizational structure for said Items. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." This is how Categories are defined in the instant specification in paragraph [0110] p. 38 "Conceptually, Categories can also be thought of as virtual Item Folders whose membership is based on the results of a specific query..., and Items that meet the conditions of this query..."). It would have been obvious to one with ordinary skill in the art to combine the computer system as disclosed in **Anglin et al.** with the Categories as disclosed in **Chang et al.** because of the advantage of having

Art Unit: 2167

the relationships stored as part of the object rather than having to execute a query to form the relationships each time. It is for this reason that one of ordinary skill in the art would have been motivated to include a plurality of Categories comprising at least one Category, wherein said plurality of Categories constitute an organizational structure for said Items.

41. Regarding claim 13, **Anglin et al.** teaches a computer system substantially as shown. **Anglin et al.** fails to teach a Category is defined by an Item property. However, **Chang et al.** teaches a Category is defined by an Item property. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." Simply by the Category being formed as the result of a query, the Item property is necessarily what defined the Category, as the data must meet the Item property in order to be a result of the query.) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in **Anglin et al.** with the disclosure of **Chang et al.** because it would be logical to use the Item property as what defines the categories, especially in the case of a query. It is for this reason that one of ordinary skill in the art would have been motivated to have a Category is defined by an Item property.

42. Regarding claim 14, **Anglin et al.** teaches a computer system substantially as shown. **Anglin et al.** fails to teach one of said plurality of Categories is defined by an Item property, and only an Item comprising the Item property for a specific Category from among said plurality of Categories can be a member of said specific Category. However, **Chang et al.** teaches one of said plurality of Categories is defined by an Item property, and only an Item comprising the Item property for a specific Category from among said plurality of Categories can be a member of said specific Category. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." Simply by the Category being formed as the result of a query, the Item property is necessarily what defined the Category, as the data must meet the Item property in order to be a result of the query. Also, by the nature of queries only returning the results that are related, the only members of the Category will be from results that comprise the Item property.) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in **Anglin et al.** with the disclosure of **Chang et al.** because it would be logical to use the Item property as what defines the categories, especially in the case of a query. It is for this reason that one of ordinary skill in the art would have been motivated to have one of said plurality of Categories is defined by an Item property, and only an Item comprising the Item property for a specific Category from among said plurality of Categories can be a member of said specific Category.

43. Regarding claim 15, **Anglin et al.** teaches a computer system substantially as shown. **Anglin et al.** fails to teach an Item comprising the Item property for one of said plurality of Categories is automatically a member of that one of said plurality of Categories. However, **Chang et al.** teaches an Item comprising the Item property for one of said plurality of Categories is automatically a member of that one of said plurality of Categories. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." By the nature of queries only returning the results that are related, the only members of the Category will be from results that comprise the Item property. Here, all of the results of the query are included in on the collection [referred to in the instant application as a category].) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in **Anglin et al.** with the disclosure of **Chang et al.** because keeping the results of the query, all of which exhibit a relationship, is useful in that the query will not have to be run again. It is for this reason that one of ordinary skill in the art would have been motivated to include an Item comprising the Item property for one of said plurality of Categories is automatically a member of that one of said plurality of Categories.

44. Regarding claim 16, **Anglin et al.** teaches a computer system substantially as shown. **Anglin et al.** fails to teach an Item comprising one or more Item properties

Art Unit: 2167

corresponding to one or more Categories of said plurality of Categories is automatically a member of each such Categories for said corresponding Item properties. However, **Chang et al.** teaches an Item comprising one or more Item properties corresponding to one or more Categories of said plurality of Categories is automatically a member of each such Categories for said corresponding Item properties. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." By the nature of queries only returning the results that are related, the only members of the Category will be from results that comprise the Item property. Here, all of the results of the query are included in on the collection [referred to in the instant application as a category].) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in **Anglin et al.** with the disclosure of **Chang et al.** because keeping the results of the query, all of which exhibit a relationship, is useful in that the query will not have to be run again. It is for this reason that one of ordinary skill in the art would have been motivated to include an Item comprising one or more Item properties corresponding to one or more Categories of said plurality of Categories is automatically a member of each such Categories for said corresponding Item properties.

45. Regarding claim 17, **Anglin et al.** teaches a computer system substantially as shown. **Anglin et al.** fails to teach each of said plurality of Categories is defined by an Item property. However, **Chang et al.** teaches each of said plurality of Categories is

Art Unit: 2167

defined by an Item property. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." Simply by the Category being formed as the result of a query, the Item property is necessarily what defined the Category, as the data must meet the Item property in order to be a result of the query.) It would have been obvious to one with ordinary skill in the art to combine the system as disclosed in **Anglin et al.** with the disclosure of **Chang et al.** because it would be logical to use the Item property as what defines the categories, especially in the case of a query. It is for this reason that one of ordinary skill in the art would have been motivated to have each of said plurality of Categories is defined by an Item property.

46. Regarding claim 18, **Anglin et al.** teaches a computer system substantially as shown. **Anglin et al.** fails to teach each of said plurality of Categories is defined by an Item property, and only Items comprising the Item property for a specific Category from among said plurality of Categories can be members of said specific Category.

However, **Chang et al.** teaches each of said plurality of Categories is defined by an Item property, and only Items comprising the Item property for a specific Category from among said plurality of Categories can be members of said specific Category. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist

Art Unit: 2167

between them.” Simply by the Category being formed as the result of a query, the Item property is necessarily what defined the Category, as the data must meet the Item property in order to be a result of the query. Also, by the nature of queries only returning the results that are related, the only members of the Category will be from results that comprise the Item property.) It would have been obvious to one with ordinary skill in the art to combine the system as disclosed in **Anglin et al.** with the disclosure of **Chang et al.** because it would be logical to use the Item property as what defines the categories, especially in the case of a query. It is for this reason that one of ordinary skill in the art would have been motivated to have each of said plurality of Categories is defined by an Item property, and only Items comprising the Item property for a specific Category from among said plurality of Categories can be members of said specific Category.

47. Regarding claim 19, **Anglin et al.** teaches a computer system substantially as shown. **Anglin et al.** fails to teach each Item comprising the Item property for one of said plurality of Categories are automatically members of that one of said plurality of Categories. However, **Chang et al.** teaches each Item comprising the Item property for one of said plurality of Categories are automatically members of that one of said plurality of Categories. (See column 13, lines 26-30 “In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them.” By the nature of queries only returning the

results that are related, the only members of the Category will be from results that comprise the Item property. Here, all of the results of the query are included in on the collection [referred to in the instant application as a category].) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in **Anglin et al.** with the disclosure of **Chang et al.** because keeping the results of the query, all of which exhibit a relationship, is useful in that the query will not have to be run again. It is for this reason that one of ordinary skill in the art would have been motivated to include each Item comprising the Item property for one of said plurality of Categories are automatically members of that one of said plurality of Categories.

48. Regarding claim 20, **Anglin et al.** teaches a computer system substantially as shown. **Anglin et al.** fails to teach all Items comprising one or more Item properties corresponding to one or more Categories of said plurality of Categories are automatically members of all such Categories for said corresponding Item properties. However, **Chang et al.** teaches all Items comprising one or more Item properties corresponding to one or more Categories of said plurality of Categories are automatically members of all such Categories for said corresponding Item properties. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." By the nature of queries only returning the results that are related, the only members of the Category will be from results that comprise the Item property.

Art Unit: 2167

Here, all of the results of the query are included in on the collection [referred to in the instant application as a category].) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in **Anglin et al.** with the disclosure of **Chang et al.** because keeping the results of the query, all of which exhibit a relationship, is useful in that the query will not have to be run again. It is for this reason that one of ordinary skill in the art would have been motivated to include all Items comprising one or more Item properties corresponding to one or more Categories of said plurality of Categories are automatically members of all such Categories for said corresponding Item properties.

49. Claim 23 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Anglin et al.** as applied to claim 21 above, and further in view of **Beauregard et al.** (US 6,438,545). **Anglin et al.** teaches a hardware/software interface system substantially as claimed. **Anglin et al.** fails to teach said Item is a fundamental unit of information manipulated by a virtual machine manager. However, **Beauregard et al.** teaches said Item is a fundamental unit of information manipulated by a virtual machine manager. (See column 13, lines 12-16 "This broad I/O capability can be provided under the Virtual Machine Manager (VMM) that is available under Win32. The VMM is an extensible operating system whose core and standard components are provided by Microsoft Corporation.") Because of the advantages provided by VMM as taught in **Beauregard et al.**, such as the broad I/O capability, it would have been obvious to one with ordinary skill in the art to combine the VMM of **Beauregard et al.** with the teaching of **Anglin et**

Art Unit: 2167

al. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item is a fundamental unit of information manipulated by a virtual machine manager.

50. Claim 24 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Anglin et al.** as applied to claim 21 above, and further in view of **Gordon and Syme** ("Typing a Multi-Language Intermediate Code" POPL '01 London, UK - Hereinafter **Gordon**).

Anglin et al. teaches a hardware/software interface system substantially as shown.

Anglin et al. fails to teach said Item is a fundamental unit of information manipulated by a Common Language Runtime. However, Hejlsberg et al. teaches said Item is a fundamental unit of information manipulated by a Common Language Runtime. (See conclusion p. 257 "One of the innovations in Microsoft's Common Language Runtime is support for typed stack pointers, for passing arguments and results by reference, for example. We presented formal typing rules and a type safety result for a substantial fragment of Common Language Runtime intermediate language. Our treatment of value types and pointer types appears to be new.") It would have been obvious to one with ordinary skill in the art to combine the teaching of **Anglin et al.** with the disclosure of **Gordon** because of the motivation of being able to pass the arguments and results by reference to have more efficient processing of the data, less transfer overhead, and to be able to handle more robust types of objects. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item is a fundamental unit of information manipulated by a Common Language Runtime.

51. Claim 25 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Anglin et al.** as applied to claim 21 above, and further in view of **Judge et al.** (US 6,430,564). **Anglin et al.** teaches a hardware/software interface system substantially as claimed. **Anglin et al.** fails to teach said Item is a fundamental unit of information manipulated by a virtual machine manager. However, **Judge et al.** teaches said Item is a fundamental unit of information manipulated by a Java Virtual Machine. (See abstract "A data manager manages global data within a Java Virtual Machine (JVM) installed and running in an embedded device. The data manager maintains a data class list that stores data class identifiers associated with each data class object currently loaded and cached in a data cache in the embedded device.") It would have been obvious to one with ordinary skill in the art to combine **Anglin et al.** with **Judge et al.** by using the JVM to allow for more diverse types of objects to be processed by the hardware/software interface system. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item is a fundamental unit of information manipulated by a Java Virtual Machine.

52. Claims 29 and 31 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Anglin et al.** as applied to claim 27 above, and further in view of **Edwards** (US Patent Application Publication 2004/0072560).

Art Unit: 2167

53. Regarding claim 29, **Anglin et al.** teaches a hardware/software interface system substantially as claimed. **Anglin et al.** fails to teach said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder. However, **Edwards** teaches said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of **Anglin et al.** with the disclosure of a recycling method of **Edwards** by simply adding the recycling method of **Edwards** to the system of **Anglin et al.** **Edwards** points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder.

54. Regarding claim 31, **Anglin et al.** teaches a hardware/software interface system substantially as claimed. **Anglin et al.** fails to teach said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder. However, **Edwards** teaches said Item, when it is a member of

only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder.) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of **Anglin et al.** with the disclosure of a recycling method of **Edwards** by simply adding the recycling method of **Edwards** to the system of **Anglin et al.** **Edwards** points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder.

55. Claims 32-36 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Anglin et al.** as applied to claim 22 above, and further in view of **Chang et al.** (US 6,578,046).

56. Regarding claim 32, **Anglin et al.** teaches a hardware/software interface system substantially as shown. **Anglin et al.** fails to teach said Item is a member of a Category. However, **Chang et al.** teaches said item is a member of a category (See

column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." This is how Categories are defined in the instant specification in paragraph [0110] p. 38 "Conceptually, Categories can also be thought of as virtual Item Folders whose membership is based on the results of a specific query..., and Items that meet the conditions of this query..."). It would have been obvious to one with ordinary skill in the art to combine the hardware/software interface system as disclosed in **Anglin et al.** with the Categories as disclosed in **Chang et al.** because of the advantage of having the relationships stored as part of the object rather than having to execute a query to form the relationships each time. It is for this reason that one of ordinary skill in the art would have been motivated to include said Item is a member of a Category.

57. Regarding claim 33, **Anglin et al.** teaches a hardware/software interface system substantially as shown. **Anglin et al.** fails to teach a Category is defined by an Item property. However, **Chang et al.** teaches said Category is defined by an Item property. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." Simply by the Category being formed as the result of a query, the Item property is necessarily what defined the Category, as the data must meet the Item property in order to be a result of the query.) It would have been obvious to one with

Art Unit: 2167

ordinary skill the art to combine the system as disclosed in **Anglin et al.** with the disclosure of **Chang et al.** because it would be logical to use the Item property as what defines the categories, especially in the case of a query. It is for this reason that one of ordinary skill in the art would have been motivated to have said Category is defined by an Item property.

58. Regarding claim 34, **Anglin et al.** teaches a hardware/software interface system substantially as shown. **Anglin et al.** fails to teach said Category is defined by an Item property, and only an Item comprising the Item property for said Category can be a member of said Category. However, **Chang et al.** teaches said Category is defined by an Item property, and only an Item comprising the Item property for said Category can be a member of said Category. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." Simply by the Category being formed as the result of a query, the Item property is necessarily what defined the Category, as the data must meet the Item property in order to be a result of the query. Also, by the nature of queries only returning the results that are related, the only members of the Category will be from results that comprise the Item property.) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in **Anglin et al.** with the disclosure of **Chang et al.** because it would be logical to use the Item property as what defines the categories, especially in the case of a query. It is

for this reason that one of ordinary skill in the art would have been motivated to have said Category is defined by an Item property, and only an Item comprising the Item property for said Category can be a member of said Category.

59. Regarding claim 35, **Anglin et al.** teaches a hardware/software interface system substantially as shown. **Anglin et al.** fails to teach an Item comprising the Item property for one of said plurality of Categories is automatically a member of that one of said plurality of Categories. However, **Chang et al.** teaches an Item comprising the Item property for one of said plurality of Categories is automatically a member of that one of said plurality of Categories. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." By the nature of queries only returning the results that are related, the only members of the Category will be from results that comprise the Item property. Here, all of the results of the query are included in on the collection [referred to in the instant application as a category].) It would have been obvious to one with ordinary skill in the art to combine the system as disclosed in **Anglin et al.** with the disclosure of **Chang et al.** because keeping the results of the query, all of which exhibit a relationship, is useful in that the query will not have to be run again. It is for this reason that one of ordinary skill in the art would have been motivated to include an Item comprising the Item property for one of said plurality of Categories is automatically a member of that one of said plurality of Categories.

60. Regarding claim 36, **Anglin et al.** teaches a hardware/software interface system substantially as shown. **Anglin et al.** fails to teach an Item comprising one or more Item properties corresponding to one or more Categories is automatically a member of each such Categories having at least one of said corresponding Item properties. However, **Chang et al.** teaches an Item comprising one or more Item properties corresponding to one or more Categories is automatically a member of each such Categories having at least one of said corresponding Item properties. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." By the nature of queries only returning the results that are related, the only members of the Category will be from results that comprise the Item property. Here, all of the results of the query are included in on the collection [referred to in the instant application as a category].) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in **Anglin et al.** with the disclosure of **Chang et al.** because keeping the results of the query, all of which exhibit a relationship, is useful in that the query will not have to be run again. It is for this reason that one of ordinary skill in the art would have been motivated to include an Item comprising one or more Item properties corresponding to one or more Categories is automatically a member of each such Categories having at least one of said corresponding Item properties.

Art Unit: 2167

61. Claims 40 and 42 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Anglin et al.** as applied to claim 38 above, and further in view of **Edwards** (US Patent Application Publication 2004/0072560).

62. Regarding claim 40, **Anglin et al.** teaches a method substantially as claimed. **Anglin et al.** fails to teach said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder. However, **Edwards** teaches said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of **Anglin et al.** with the disclosure of a recycling method of **Edwards** by simply adding the recycling method of **Edwards** to the system of **Anglin et al.** **Edwards** points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder.

Art Unit: 2167

63. Regarding claim 42, **Anglin et al.** teaches a method substantially as claimed. **Anglin et al.** fails to teach said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder. However, **Edwards** teaches said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder.) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of **Anglin et al.** with the disclosure of a recycling method of **Edwards** by simply adding the recycling method of **Edwards** to the system of **Anglin et al.** **Edwards** points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder.

64. Claims 46 and 48 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Anglin et al.** as applied to claim 44 above, and further in view of **Edwards** (US Patent Application Publication 2004/0072560).

65. Regarding claim 46, **Anglin et al.** teaches a computer readable medium substantially as claimed. **Anglin et al.** fails to teach said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder. However, **Edwards** teaches said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of **Anglin et al.** with the disclosure of a recycling method of **Edwards** by simply adding the recycling method of **Edwards** to the system of **Anglin et al.** **Edwards** points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder.

66. Regarding claim 48, **Anglin et al.** teaches a computer readable-medium substantially as claimed. **Anglin et al.** fails to teach said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member

Art Unit: 2167

of a default Item Folder. However, **Edwards** teaches said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder.) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of **Anglin et al.** with the disclosure of a recycling method of **Edwards** by simply adding the recycling method of **Edwards** to the system of **Anglin et al.** **Edwards** points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder.

67. Claims 52 and 54 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Anglin et al.** as applied to claim 50 above, and further in view of **Edwards** (US Patent Application Publication 2004/0072560).

68. Regarding claim 52, **Anglin et al.** teaches a computer readable medium substantially as claimed. **Anglin et al.** fails to teach said Item, when it no longer

Art Unit: 2167

belongs to any Item Folder, automatically becomes a member of a default Item Folder. However, **Edwards** teaches said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of **Anglin et al.** with the disclosure of a recycling method of **Edwards** by simply adding the recycling method of **Edwards** to the system of **Anglin et al.** **Edwards** points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder.

69. Regarding claim 54, **Anglin et al.** teaches a computer-readable medium substantially as claimed. **Anglin et al.** fails to teach said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder. However, **Edwards** teaches said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to

Art Unit: 2167

'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder.) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of **Anglin et al.** with the disclosure of a recycling method of **Edwards** by simply adding the recycling method of **Edwards** to the system of **Anglin et al.** **Edwards** points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder.

70. Claims 58 and 60 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Anglin et al.** as applied to claim 56 above, and further in view of **Edwards** (US Patent Application Publication 2004/0072560).

71. Regarding claim 58, **Anglin et al.** teaches a computer-readable medium substantially as claimed. **Anglin et al.** fails to teach an Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder. However, **Edwards** teaches an Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The

Art Unit: 2167

synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of **Anglin et al.** with the disclosure of a recycling method of **Edwards** by simply adding the recycling method of **Edwards** to the system of **Anglin et al.** **Edwards** points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have an Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder.

72. Regarding claim 60, **Anglin et al.** teaches a computer-readable medium substantially as claimed. **Anglin et al.** fails to teach said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder. However, **Edwards** teaches said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved

Art Unit: 2167

if required." In the instant application, the area in which the item is stored is called the default Item Folder.) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of **Anglin et al.** with the disclosure of a recycling method of **Edwards** by simply adding the recycling method of **Edwards** to the system of **Anglin et al.** **Edwards** points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder.

Conclusion

73. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Chou et al. (US 6,370,541) teaches relationships between groups.

Bolik et al. (US Patent Application Publication 2003/0196052) is similar to **Anglin et al.** but refers to Backup databases

Govindarajan et al. (US 6,006,234) teaches logical grouping of database objects.

Jeremy Singer, "JVM versus CLR: A Comparative Study", 16-18 June 2003, PPPJ 2003, Kilkenny City, Ireland, page 167.

Art Unit: 2167

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Dennis L. Vautrot whose telephone number is 571-272-2184. The examiner can normally be reached on Monday-Friday 8:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cunningham can be reached on 571-272-7079. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Dv
21 April 2006


JOHN R. COTTINGHAM
PRIMARY EXAMINER

